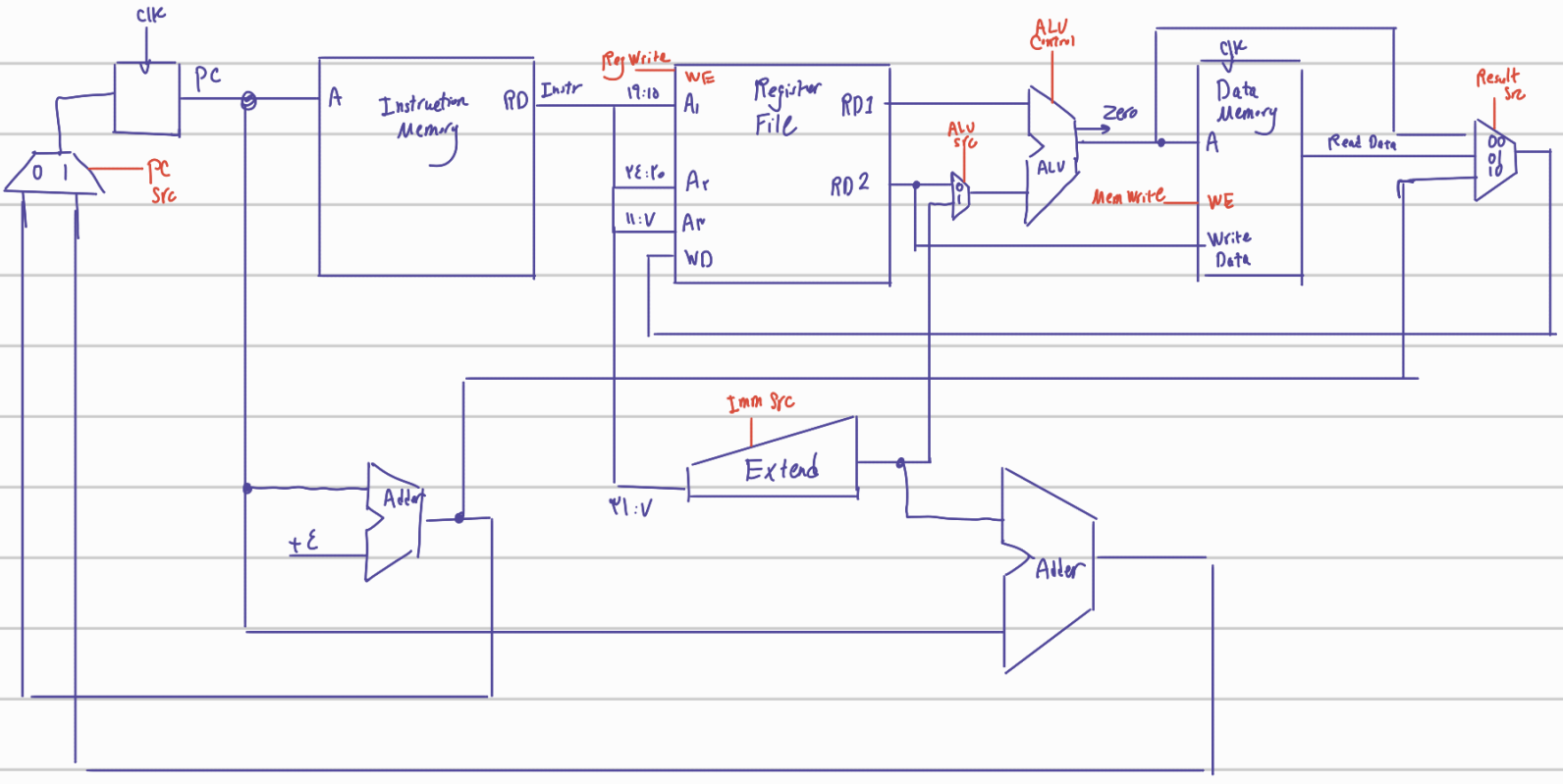
CA 2 : RISC-V Single Cycle

Mohammad Amin Rashid 810102454

Mohammad Amin Jangi 810102429

Data path :



R Type : add sub and or slt

I Type : lw addi ori slti Jalr

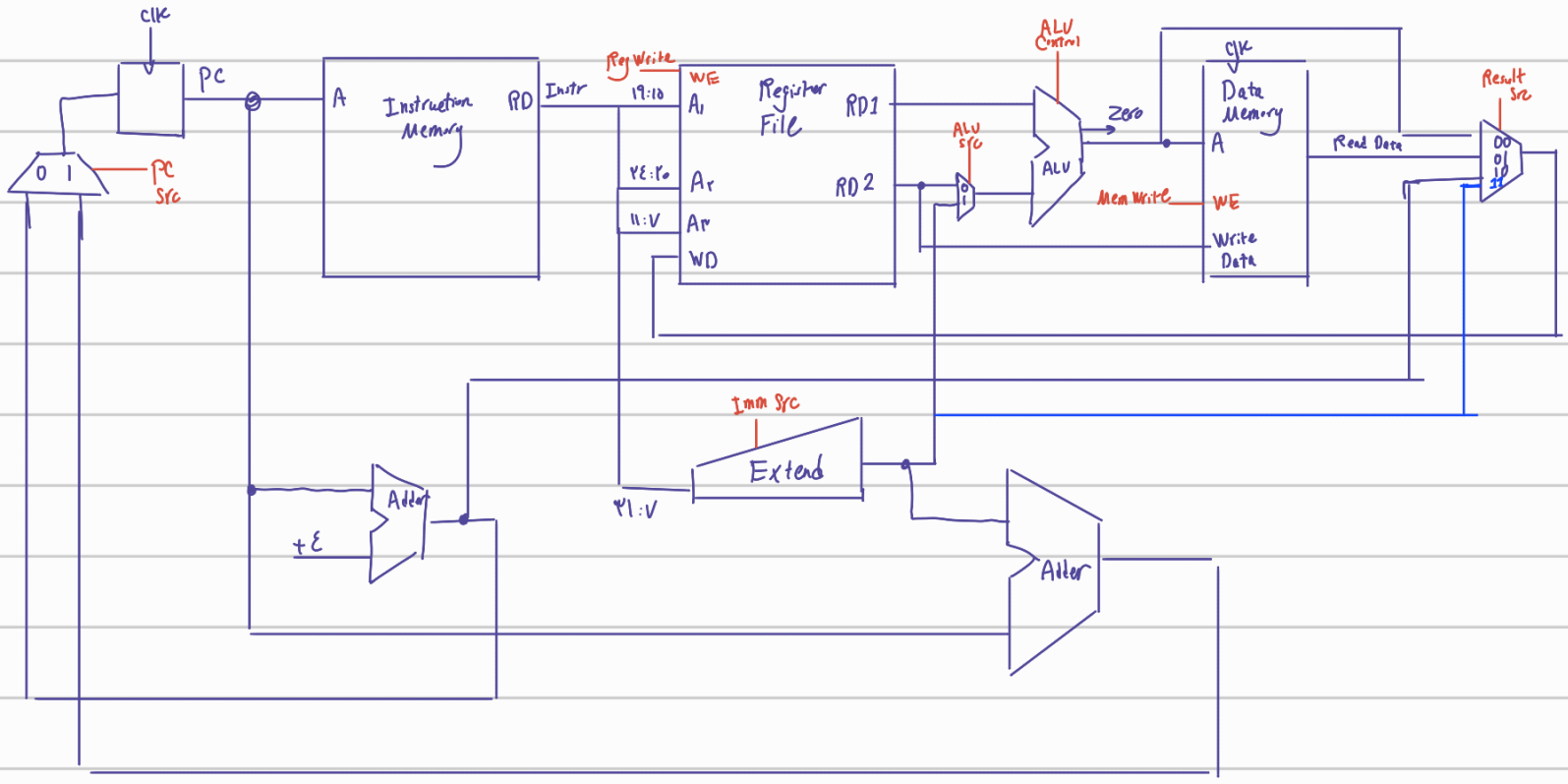
S Type : sw

B Type : beq bne

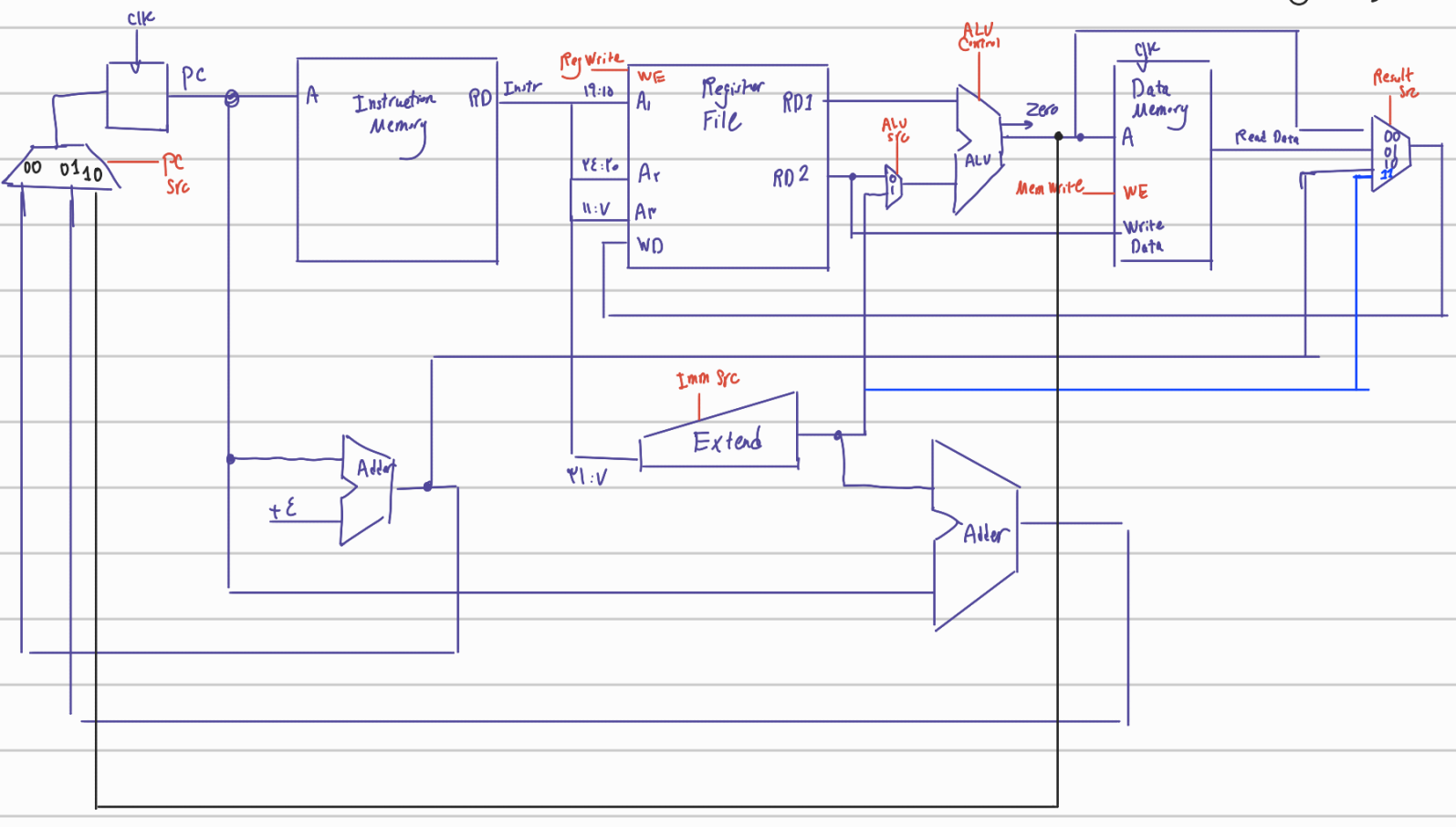
U Type : lui

J Type : Jal

دستور Lui :



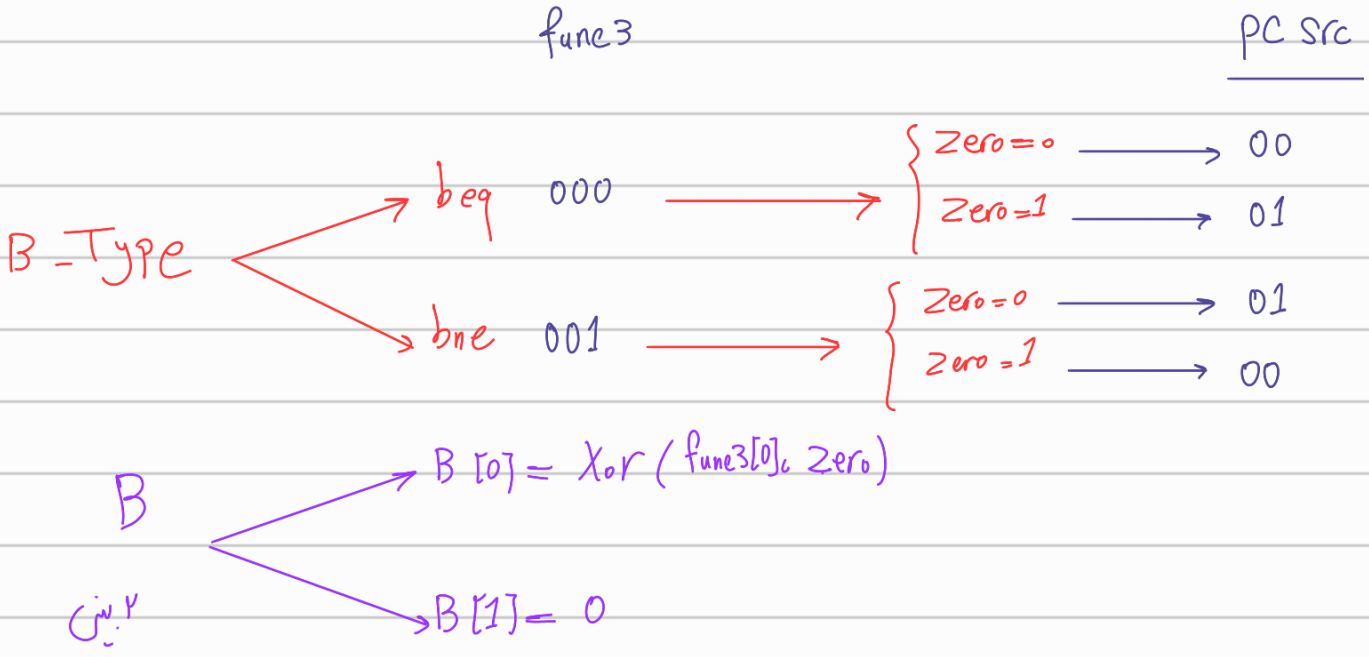
دستور Jalr :



Controller:

	(Vbit) opc	Reg Write	Imm Src	ALU Src	Mem Write	Result Src	ALU opc	PC Src
RType (add sub and or slt)	0110011	1	xxx	0	0	00	10	00
I Type (addi ori slli)	0010011	1	000	1	0	00	10	00
I Type (lw)	0000011	1	000	1	0	01	00	00
I Type (Jalr)	1100111	1	000	1	0	10	00	10
S Type (sw)	0100011	0	001	1	1	xx	00	00
B Type (beq bne)	1100011	0	010	0	0	xx	01	B
U Type (lui)	0110111	1	100	x	0	11	xx	00
J Type (Jal)	1101111	1	011	x	0	10	xx	01

ALU opc	f_3	f_7	ALU Control
00	X	X	000 (+)
01	X	X	001 (-)
10	000	$\neq 0100000$	000 (+)
	000	0100000	001 (-)
	111		010 (&)
	110		011 (|)
	010		101 (slt)



32-bit Instruction Formats

	31–27	26–25	24–20	19–15	14–12	11–7	6–0
R	funct7		rs2	rs1	funct3	rd	opcode
I	imm[11:0]			rs1	funct3	rd	opcode
S	imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode
B	imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode
U	imm[31:12]					rd	opcode
J	imm[20|10:1|11|19:12]					rd	opcode

```
Jal ra, 8
sw s0, 84(zero)

add s0, zero, zero
add s1, zero, zero
add s4, zero, zero
add s5, zero, zero
Loop: slti t1, s1, 20
beq t1, zero, EndLoop
lw s3, 0(s5)
slt t3, s3, s0
bne t3, zero, EndIf
add s0, zero, s3
add s4, zero, s1

EndIf: addi s5, s5, +4
add s1, s1, +1

Jal zero, Loop

EndLoop:
Jalr zero, ra, 0
```

s0 ← max
s1 ← i
s4 ← index of max
s5 ← &i

Register	ABI Name
x0	zero
x1	ra
x2	sp
x3	gp
x4	tp
x5-7	t0-2
x8	s0/fp
x9	s1
x10-11	a0-1
x12-17	a2-7
x18-27	s2-11
x28-31	t3-t6

```
008000ef
04802a23
00000433
000004b3
00000933
00000a33
0144a313
02030263
000a2983
0089a3b3
00039663
01300433
00900933
00148493
004a0a13
fddff06f
00008067
```

```
1A000000 → 26
3F000000 → 63
2B000000 → 43
7C000000 → 124
05000000 → 5
88000000 → 136
19000000 → 25
64000000 → 100
31000000 → 49
0E000000 → 14
55000000 → 85
22000000 → 34
AF000000 → 175
11000000 → 17
40000000 → 64
9B000000 → 155
6A000000 → 106
03000000 → 3
FE000000 → 254
D4000000 → 212
```